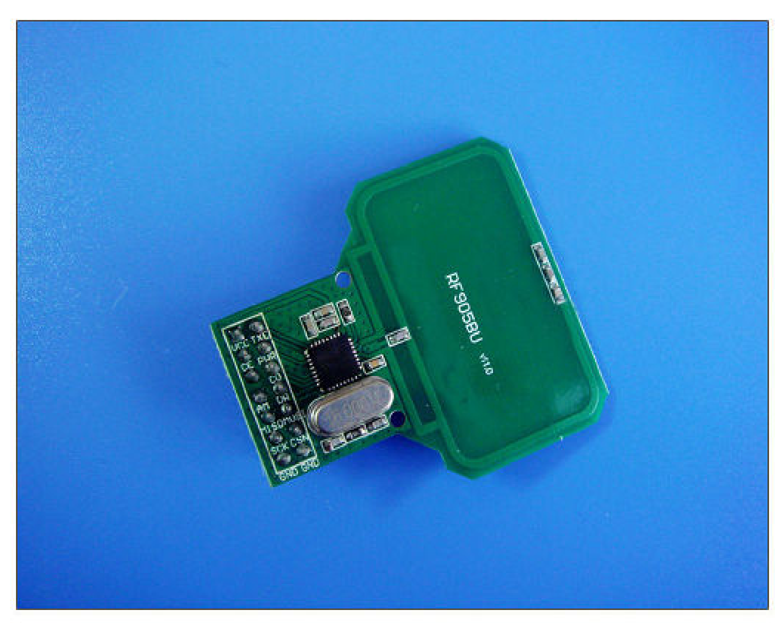
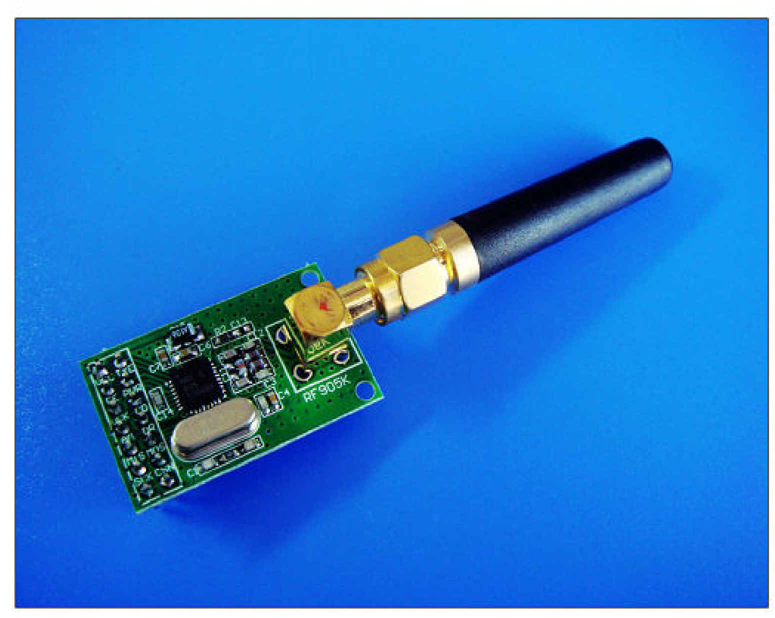
nRF905 无线模块
用户手册





联系电话：13704018223　陈 工　　E-mail: chj_006@sina.com
在线咨询：QQ:35625400　474882985　　MSN:1188mm88@hotmail.com

产品概述

nRF905 是挪威 Nordic 公司推出的一款单片射频发射器芯片，采用 32 引脚 5mm×5mm QFN 封装，工作于 433、868、915MHz 3 个 ISM（工业、科学和医学）频道，其中国内 433 频段可以免费使用。nRF905 由频率合成器、接收解调器、功率放大器、晶体振荡器、调制器等功能组成,不需要外加声表滤波器也可以有良好的通信效果。nRF905 使用 SPI 接口可以和任何 MCU 进行通信，其中地址、输出功率和通信频道可通过程序进行配置，所以可以用于多机通信。nRF905 融合了 ShockBurstTM 技术，可以自动处理数据包字头,且内置 CRC 校验功能,确保数据可靠传输。nRF905 功耗很低,在以-10dBm 的功率发射时，工作电流也只有 11mA;而对应接收机的工作电流只有 12.5 mA，芯片可以软件设置空闲模式、关机模式，易于节能设计。适合工业数据采集、无线报警及安全系统等诸多领用。

基本特点

(1) 433Mhz 开放 ISM 频段免许可证使用；
(2) 最高工作速率 50kbps,通信距离可达 300 米左右；
(3) 高效 GFSK 调制，抗干扰能力强，特别适合工业控制场合；
(4) 工作频率可软件设置，满足多点通信和跳频通信需要；
(5) 内置硬件 8、16 位 CRC 检错和点对多点通信地址控制；

(6) 低功耗 1.9 - 3.6V 工作，待机模式下状态仅为 2.5uA ；

(7) 收发模式切换时间仅 650us

(8) SPI 编程接口，可软件设置地址，地址多达 2 的 32 次方

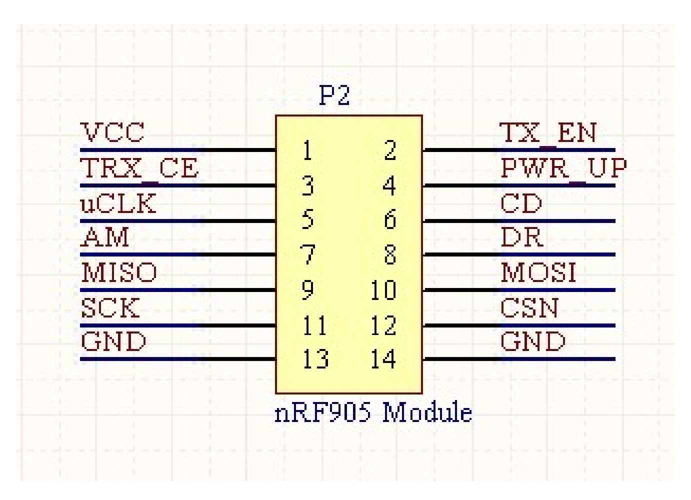
(9) 集成地址匹配、载波侦听、收发完成状态指示功能

(10) TX Mode：在+10dBm 情况下，电流为 30mA；RX Mode：12.2mA

(11) 标准 2.54mm DIP 间距接口，便于嵌入式应用

同时，为便于用户开发，我们提供配套评估套件，为产品开发保驾护航，使无线应用开发大大加速，并避免不必要的误区。

引脚接口说明



引脚功能说明

管脚	名称	管脚功能	说明
1	VCC	电源	电源+1.9-3.6V DC
2	TX_EN	数字输入	TX_EN=1 TX模式，TX_EN=0 RX模式
3	TRX_CE	数字输入	使能芯片发射或接收
4	PWR_UP	数字输入	芯片上电

5	uCLK	时钟输出	模块中该引脚废弃，便于向后兼容
6	CD	数字输出	载波检测
7	AM	数字输出	地址匹配
8	DR	数字输出	接收或发射数据完成
9	MISO	SPI 接口	SPI 输出
10	MOSI	SPI 接口	SPI 输入
11	SCK	SPI 时钟	SPI 时钟
12	CSN	SPI 使能	SPI 使能
13	GND	数字地	接地
14	GND	数字地	接地

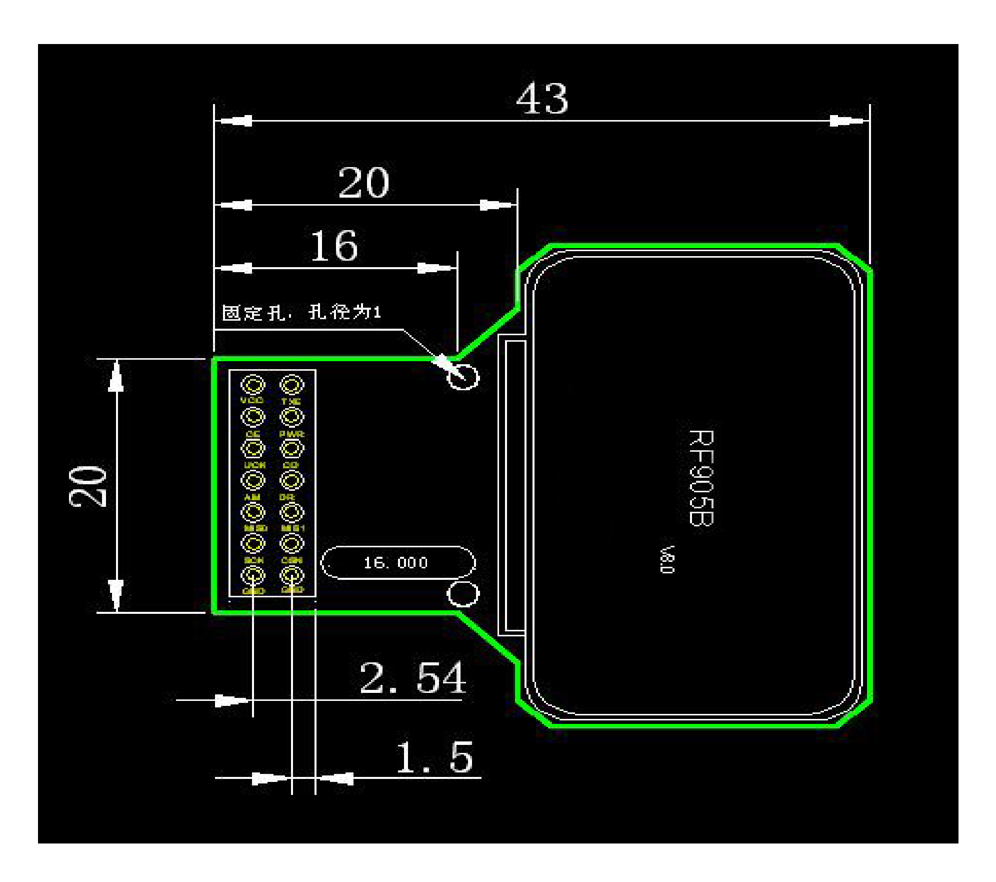
1.VCC 引脚的电压范围为1.9-3.6V 之间，不能在这个区间之外，如超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右；

2.硬件没有集成SPI功能的单片机也可以控制本模块，用普通单片IO口模拟 SPI 时序进行读写操作即可；

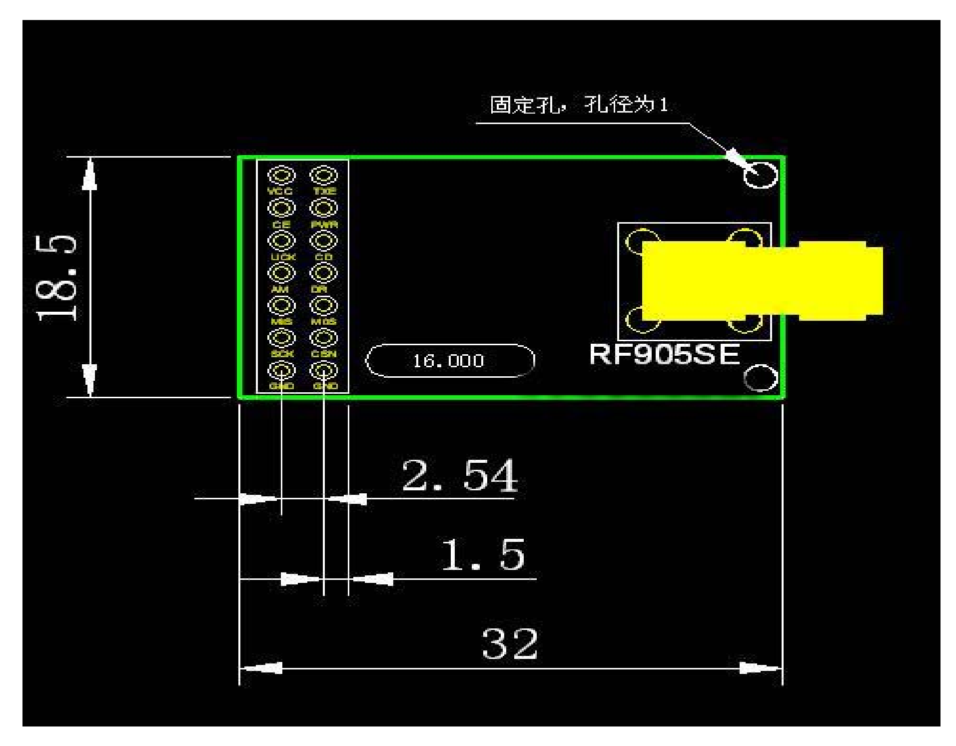
3. 模块接口采用标准2.54mmDIP插针，13 脚、14 脚为接地脚,需要和系统电路的逻辑地连接起来；

4.与 51 系列单片机 P0 口连接时候，需要加 10K 的上拉电阻,与其余口连接不需要。其他系列的5V单片机，**如AVR、PIC，请参考该系列单片机 IO 口输出电流大小，如果超过 10mA，需要串联2-5K电阻分压**，否则容易烧毁模块！如果是 3.3V 的MCU，可以直接和IO口连接。

模块尺寸（单位：mm）



RF905B 尺寸图（PCB 差分环形天线）



RF905SE 尺寸图（SMA 天线接口）

模块电气参数

参数	数值	单位
工作电压	1.9-3.6	V
最大发射功率	10	dBm
最大数据传输率曼切斯特编码	50	kbps
输出功率为-10 dBm 时工作电流	9	mA
接收模式时工作电流	12.5	mA
温度范围	-40 to +85	
典型灵敏度	-100	dBm
POWERDOWN 模式时工作电流	2.5	uA

nRF905 模块工作电压对最大发射增益影响

工作电压(模块VCC供电电压)	模块最大发射增益(dBm)
+3.3V	+7.3dBm
+3.6V	+10dBm

工作方式

nRF905一共有四种工作模式，其中有两种正常RX/TX 模式和两种节电模式。

正常模式

ShockBurst RX　　　ShockBurst TX

节电模式

掉电和SPI编程　　　STANDBY 和 SPI编程

nRF905 工作模式由TRX_CE、TX_EN、PWR_UP 的设置来设定。

PWR_UP	TRX_CE	TX_EN	工作模式
0	X	X	掉电和SPI 编程
1	0	X	Standby 和SPI 编程
1	1	0	ShockBurst RX
1	1	1	ShockBurst TX

ShockBurst 模式

ShockBurstTM 收发模式下，使用片内的先入先出堆栈区，数据低速从微控制器送入，但高速发射，这样可以尽量节能，因此，使用低速的微控制器也能得到很高的射频数据发射速率。与射频协议相关的所有高速信号处理都在片内进行，这种做法有三大好处：尽量节能；低的系统费用(低速微处理器也能进行高速射频发射)；数据在空中停留时间短，抗干扰性高。ShockBurstTM 技

术同时也减小了整个系统的平均工作电流。

在 ShockBurstTM 收发模式下,nRF905 自动处理字头和 CRC 校验码。在接收数据时，自动把字头和 CRC 校验码移去。在发送数据时，自动加上字头和 CRC 校验码，当发送过程完成后，DR 引脚通知微处理器数据发射完毕。

ShockBurst TX 发送流程

nRF905 数据发送流程分以下几步：

A. 当微控制器有数据要发送时，通过 SPI 接口，按时序把接收机的地址和要发送的数据送传给 nRF905，SPI 接口的速率在通信协议和器件配置时确定；

B. 微控制器置高 TRX_CE 和 TX_EN,激发 nRF905 的 ShockBurstTM 发送模式；

C. nRF905 的 ShockBurstTM 发送：

(1) 射频寄存器自动开启；

(2) 数据打包(加字头和 CRC 校验码)；

(3) 发送数据包；

(4) 当数据发送完成，数据准备好引脚被置高；

D. AUTO_RETRAN 被置高，RF905 不断重发，直到 TRX_CE 被置低；

E. 当 TRX_CE 被置低，RF905 发送过程完成，自动进入空闲模式。

注意：ShockBurstTM 工作模式保证，一旦发送数据的过程开始，无论 TRX_EN 和 TX_EN 引脚是高或低，发送过程都会被处

理完。只有在前一个数据包被发送完毕，RF905 才能接受下一个发送数据包。

ShockBurst RX 接收流程

nRF905 数据接收流程分以下几步：

A. 当 TRX_CE 为高、TX_EN 为低时，nRF905 进入 ShockBurstTM 接收模式；

B. 650us 后,nRF905 不断监测，等待接收数据；

C. 当 nRF905 检测到同一频段的载波时，载波检测引脚被置高；

D. 当接收到一个相匹配的地址，AM 引脚被置高；

E. 当一个正确的数据包接收完毕,nRF905 自动移去字头、地址和 CRC 校验位，然后把 DR 引脚置高

F. 微控制器把 TRX_CE 置低，nRF905 进入空闲模式；

G. 微控制器通过 SPI 口，以一定的速率把数据移到微控制器内；

H. 当所有的数据接收完毕，nRF905 把 DR 引脚和 AM 引脚置低；

I. nRF905 此时可以进入 ShockBurstTM 接收模式、ShockBurstTM 发送模式或关机模式。

当正在接收一个数据包时，TRX_CE 或 TX_EN 引脚的状态发生改变，nRF905 立即把其工作模式改变，数据包则丢失。当微处理器接到 AM 引脚的信号之后， 其就知道 nRF905 正在接收数据包，其可以决定是让 nRF905 继续接收该数据包还是进入另一个工作模式。

节能模式

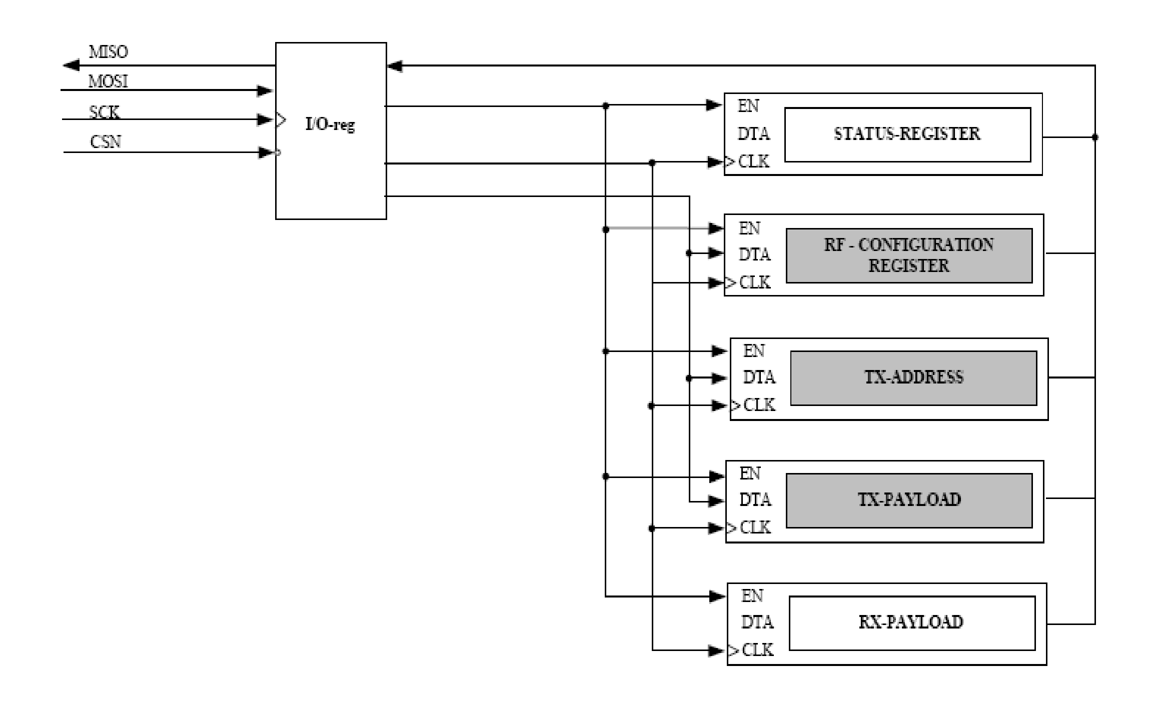
nRF905 的节能模式包括关机模式和节能模式。在掉电模式，nRF905 的工作电流最小，一般为 2.5uA。进入关机模式后，nRF905 保持配置字中的内容，但不会接收或发送任何数据。在 Standby 模式下，RF905 内部的部分晶体振荡器处于工作状态。该模式有利于减小工作电流，其从空闲模式到发送模式或接收模式的启动时间最短。

nRF905 模块参数设置

所有配置字都是通过 SPI 接口送给 RF905。SIP 接口的工作方式可通过 SPI 指令进行设置。当 RF905 处于空闲模式或关机模式时，SPI 接口可以保持在工作状态。

寄存器配置由状态寄存器、射频配置寄存器、发送地址寄存器、发送数据寄存器和接收数据寄存器 5 个寄存器组成。状态寄存器包含数据准备好引脚状态信息和地址匹配引脚状态信息；射频配置寄存器包含收发器配置信息，如频率和输出功能等；发送地址寄存器包含接收机的地址和数据的字节数；发送数据寄存器包含待发送的数据包的信息，如字节数等；接收数据寄存器包含要接收的数据的字节数等信息。

SPI接口由5个内部寄存器组成执行寄存器的回读模式来确认寄存器的内容



SPI接口寄存器示意图

1.状态寄存器Status-Register

寄存器包含数据就绪DR 和地址匹配AM 状态

2.配置寄存器RF-Configuration Register

寄存器包含收发器的频率,输出功率等配置信息

3.发送地址TX-Address

寄存器包含目标器件地址字节长度由配置寄存器设置

4.发送有效数据TX-Payload

寄存器包含发送的有效ShockBurst 数据包数据字节长度由配置寄存器设置

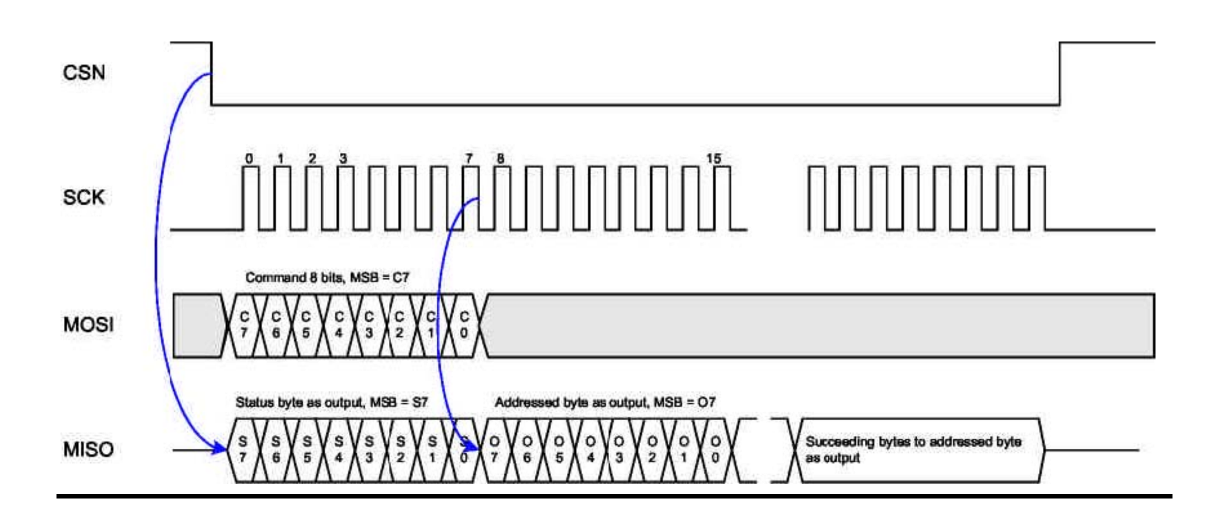
5.接收有效数据TX-Payload

寄存器包含接收的有效ShockBurst 数据包数据字节长度由

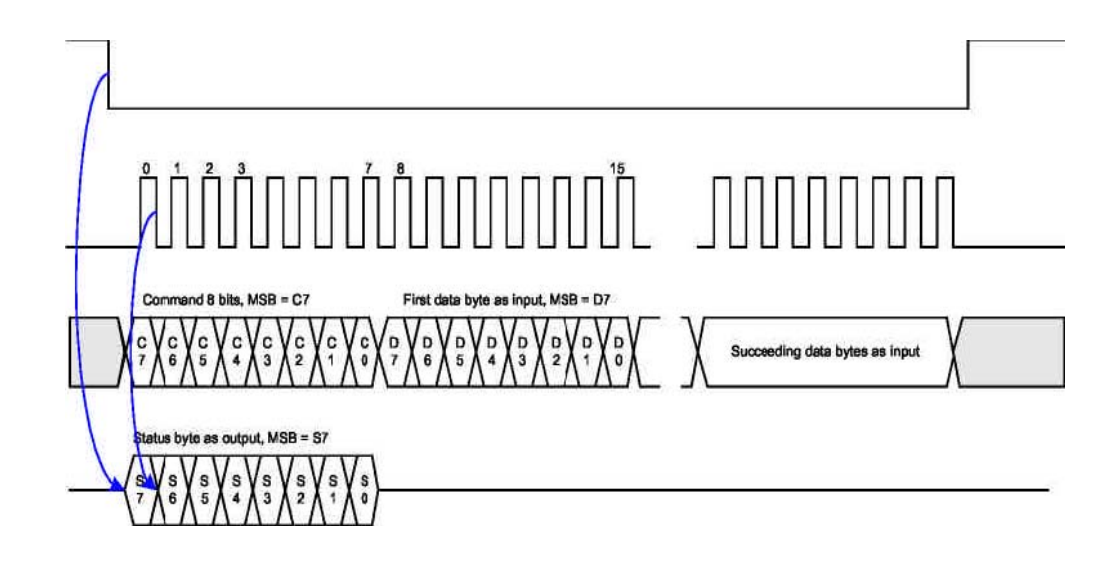
配置寄存器设置在寄存器中的有效数据由数据准备就绪DR 指示

SPI 读写操作时序

SPI 读操作时序



SPI 写操作时序



PARAMETER	SYMBOL	MIN	MAX	UNITS
Data to SCK Setup	Tdc	5		ns
SCK to Data Hold	Tdh	5		ns
CSN to Data Valid	Tcsd		45	ns
SCK to Data Valid	Tcd		45	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Tsck	DC	10	MHz
SCK Rise and Fall	Tr,Tf		100	ns
CSN to SCK Setup	Tcc	5		ns
SCK to CSN Hold	Tcch	5		ns
CSN Inactive time	Tcwh	500		ns
CSN to Output High Z	Tcdz		45	ns

Table 8 SPI timing parameters (C_{load} = 10pF).

nRF905 模式切换时间

nRF905 timing	Max.
PWR_DWN ➔ ST_BY mode	3 ms
STBY ➔ TX ShockBurst™	650 µs
STBY ➔ RX ShockBurst™	650 µs
RX ShockBurst™ ➔ TX ShockBurst™	550 [1]µs
TX ShockBurst™ ➔ RX ShockBurst™	550 [1]µs

SPI 读写操作指令解析

当CSN 为低时，SPI接口开始等待一条指令。任何一条新指令均由CSN 的由高到低的转换开始。用于SPI 接口的有用命令见下表：请认真阅读一下一系列表格内容介绍，是编程关键

指令名称	指令格式	操作
W_CONFIG （WC）	0000AAAA	写配置寄存器AAAA 指出写操作的开始字节字节数量取决于AAAA 指出的开始地址
R_CONFIG （RC）	0001AAAA	读配置寄存器AAAA 指出读操作的开始字节字节数量取决于AAAA 指出的开始地址
W_TX_PAYLOA D （WTP）	00100000	写TX 有效数据1-32 字节写操作全部从字节0 开始

R_TX_PAYLOA D (RTP)	00100001	读TX 有效数据1-32 字节读操作全部从字节0 开始
W_TX_ADDRES S (WTA)	00100010	写TX 地址1-4 字节写操作全部从字节0 开始
R_TX_ADDRES S (RTA)	00100011	读TX 地址1-4 字节读操作全部从字节0 开始
R_RX_PAYLOA D (RRP)	00100100	读RX 有效数据1-32 字节读操作全部从字节0 开始
CHANNEL_CON FIG (CC)	1000pphc cccccccc	快速设置配置寄存器中CH_NO HFREQ_PLL 和PA_PWR 的专用命令CH_NO=ccccccccc HFREQ_PLL=h PA_PWR=pp

配置寄存器分析

参数	位宽	说明
CH_NO	9	同HFREQ_PLL 一起设置中心频率默认值 =001101100b=180d FRF= 422.4+ CH_NOd/10 *(1+ HFREQ_PLLd)MHZ
HFREQ_ PLL	1	设置PLL 在433 或868/915MHZ 模式默认值=0 0 -器件工作在433MHZ 频段1 -器件工作在868/915MHZ 频段
PA_PWR	2	输出功率默认值=00 00 -10dBm 01 -2dBm 10 +6dBm 11 +10dBm
RX_RED_ PWR	1	降低接收模式电流消耗至1.6mA 灵敏度降低默认值=0 0 -正常模式1 -低功耗模式
AUTO_ RETRAN	1	重发数据如果TX 寄存器的TRX_CE 和TX_EN 被设置为高默认值=0 0 -不重发数据1 -重发数据包
RX_AWF	3	RX 地址宽度默认值=100 001 -1 字节RX 地址宽度100 -4 字节RX 地址宽度
TX_AWF	3	TX 地址宽度默认值=100 001 -1 字节TX 地址宽度100 -4 字节TX 地址宽度
RX_PW	6	RX 接收有效数据宽度默认值=100000 000001 -1 字节RX 有效数据宽度000010 -2 字节RX 有效数据宽度100000 -32 字节RX 有效数据宽度
TX_PW	6	TX 有效数据宽度默认值=100000 000001 -1 字节TX 有效数据宽度000010 -2 字节TX 有效数据宽度100000 -32 字节TX 有效数据宽度
RX_ ADDRESS	32	RX 地址使用字节依赖于RX_AFW 默认值=E7E7E7E7h

UP_CLK_ FREQ	2	输出时钟频率默认值=11 00 -4MHZ 01 -2MHZ 10 -1MHZ 11 -500KHZ
UP_CLK_EN	1	输出时钟使能默认值=1 0 -没有外部时钟1 -外部时钟信号使能
XOF	3	晶体振荡器频率必须依据外部晶体的标称频率设置默认值=100 000 -4MHZ 001 -8MHZ 010 -12MHZ 011 -16MHZ 100 -20MHZ
CRC_EN	1	CRC 校验允许默认值=1 0 -不允许1 -允许
CRC_ MODE	1	CRC 模式默认值=1 0 -8 位CRC 校验位1 -16 位CRC 校验位

寄存器配置顺序

RF-Configuration-Register(R/W)		
字节#	内容位[7 0] MSB=BIT[7]	初始化值
0	Bit[7 0]	0110_1100
1	Bit[7:6]没用AUTO_RETRAN RX_RED_PWR PA_PWR[1:0] HFREQ_PLL CH_NO[8]	0000_0000
2	Bit[7] 没用TX_AFW[2:0] Bit[3] 没用 RX_AFW[2:0]	0100_0100
3	Bit[7:6]没用RX_PWR[5:0]	0010_0000
4	Bit[7:6]没用TX_PWR[5:0]	0010_0000
5	RX 地址0 字节	E7
6	RX 地址1 字节	E7
7	RX 地址2 字节	E7
8	RX 地址3 字节	E7
9	CRC_ 模式 CRC 校验允许X OF[2:0] UP_CLK_EN UP_CLK_FREQ[1:0]	1110_0111

TX_PAYLOAD(R/W)		
字节#	内容位[7 0] MSB=BIT[7]	初始化值
0	TX_PAYLOAD[7:0]	X
1	TX_PAYLOAD[15:8]	X
		X
		X
30	TX_PAYLOAD[247:240]	X
31	TX_PAYLOAD[255:248]	X

TX_ADDRESS(R/W)		
字节#	内容位[7 0] MSB=BIT[7]	初始化值

0	TX_ADDRESS[7:0]	E7
1	TX_ ADDRESS [15:8]	E7
2	TX_ ADDRESS [23:16]	E7
3	TX_ ADDRESS [31:24]	E7

RX_PAYLOAD(R)		
字节#	内容位[7 0] MSB=BIT[7]	初始化值
0	RX_PAYLOAD[7:0]	X
1	RX_PAYLOAD[15:8]	X
		X
		X
30	RX_PAYLOAD[247:240]	X
31	RX_PAYLOAD[255:248]	X

STATUS_REGISTER(R)		
字节#	内容位[7 0] MSB=BIT[7]	初始化值
0	AM bit[6] 没用DR bit[4:0] 没用	E7

字节 0：（第一字节）

[7:0] CH_NO[7:0]：

连同字节 1 的 CH_NO[8]和 HFREQ_PLL 控制 905 的载波频段

以下参考设置：

Frequency	HFREQ_PLL	CH_NO
430.0 MHz	[0]	[001001100]
433.1 MHz	[0]	[001101011]
433.2 MHz	[0]	[001101100]
434.7 MHz	[0]	[001111011]
862.0 MHz	[1]	[001010110]
868.2 MHz	[1]	[001110101]
868.4 MHz	[1]	[001110110]
869.8 MHz	[1]	[001111101]

902.2 MHz	[1]	[100011111]
902.4 MHz	[1]	[100100000]
927.8 MHz	[1]	[110011111]

载波频率的计算公式：

$$f_{OP} = (422.4 + (CH_NO/10)) \cdot (1 + HFREQ_PLL)\ MHz$$

字节 1：

[0] CH_NO [8] ：参见字节 0

[1] HFREQ_PLL ：

0 – 器件工作在 433MHZ 频段

1 – 期间工作在 868/915MHZ 频段

[3:2] PA_PWR ：

输出功率

00 -10dBm （默认）

01 -2dBm

10 +6dBm

11 +10dBm

[4] RX_RED_PWR ：

降低接收模式电流消耗至 1.6mA，灵敏度降低。

0 – 正常模式 （默认）

1 – 低功耗模式

[5] AUTO_RETRAN：

自动重发 TX 寄存器中的数据包，如果 TRX_CE 和 TX_EN 被设置为高。

0 – 不重发数据包 （默认）

1 – 自动重发数据包

[7:6] 保留

字节 2

[2:0] RX_AWF [2:0] ：（一般设置成 4 字节更可靠）

RX 地址宽度

001 – 1 字节 RX 地址宽度 （默认）

100 – 4 字节 RX 地址宽度

[3] 保留

[6:4] TX_AWF [2:0] ：（一般设置成 4 字节更可靠）

TX 地址宽度

001 – 1 字节 TX 地址宽度

100 – 4 字节 TX 地址宽度

[7] 保留

字节 3

[5:0] RX_PW [5:0] ：（可以设置称 1、2、4、8、16、32 字节）

RX 接收有效数据宽度

000001 – 1 字节 RX 有效数据宽度

000010 – 2 字节 RX 有效数据宽度

……

100000 – 32 字节 RX 有效数据宽度

[7:6] 保留

字节 4

[5:0] TX_PW [5:0] ：（可以设置称 1、2、4、8、16、32 字节）

TX 发送有效数据宽度

000001 – 1 字节 TX 有效数据宽度

000010 – 2 字节 TX 有效数据宽度

……

100000 – 32 字节 TX 有效数据宽度

[7:6] 保留

字节 5： RX 地址 0 字节

字节 6： RX 地址 1 字节

字节 7： RX 地址 2 字节

字节 8： RX 地址 3 字节

字节 9

[1:0] UP_CLK_FREQ [1:0]：

输出时钟频率

00 – 4MHZ

01 – 2MHZ

10 – 1MHZ

11 – 500KHZ

[2] UP_CLK_EN ：（一般禁止时钟输出）

输出时钟使能

0 - 没有外部时钟

1 - 外部时钟信号使能 （默认）

[5:3] XOF [2:0] ：

晶体振荡器频率，必须依据外部晶体的标称频率设置

000 - 4MHZ

001 - 8MHZ

010 - 12MHZ

011 - 16MHZ（模块标配为 16MHZ）

100 - 20MHZ

[6] CRC_EN ：（一般都使能，提高数据传输可靠性）

CRC 校验允许

0 - 部允许

1 - 允许 （默认）

[7] CRC_MODE ：

CRC 模式

0 - 8 位 CRC 校验位

1 -16 位 CRC 校验位 （默认）

nRF905 编程指南

使用RF905模块无需掌握任何专业无线或高频方面的理论，读者只需要具备一定的C语言程序基础即可。本文档没有涉及到的问题，读者可以参考nRF905官方手册或向我们寻求技术支持。同时，为便于用户开发，我们提供配套评估套件，为产品开发保驾护航，使无线应用开发大大加速，并避免不必要的误区。

nRF905 代码解析

```
unsigned char idata RFConf[11]=
{
  0x00,                          //配置命令 WC//
  0x4c,                          //CH_NO,配置频段在 430MHZ
  0x0c,                          //输出功率为 10db,不重发，节电为正常模式
  0x44,                          //地址宽度设置，为 4 字节
  0x04,0x04,                     //接收发送有效数据长度为 4 字节
  0xCC,0xCC,0xCC,0xCC,           //接收地址
  0x58,            //CRC 充许，8 位 CRC 校验，外部时钟信号不使能，16M 晶振
};
```

注：对于频段设置参数 CH_NO，在我们提供的范例程序中 CH_NO[7:0]的值为 0x4c。我们不建议各位用户使用其他数值，因为我们的模块在硬件上只适应 430MHz 左右的频率，为了达到最好的效果，软件参数上应当与硬件匹配，否则会影响通讯距离。

nRF905 通过 SPI 接口与单片机通讯，必须首先了解 SPI 接口。

SPI 外围串行接口由四条线构成：

MOSI 主机输出从机输入　　（主机写操作）

MISO 主机输入从机输出　　（主机读操作）

SCK 串行时钟信号，由主机控制

CSN 片选信号，低电平有效

nRF905 SPI 接口指令的宏定义

```
//（以下操作全部从对应寄存器的字节 0 开始）
#define WC     0x00
// 写配置寄存器（RF-Configuration Register）
#define RC     0x10
// 读配置寄存器（RF-Configuration Register）
#define WTP        0x20
// 向 TX-Payload 寄存器写入发送有效数据
#define RTP        0x21
// 从 TX-Payload 寄存器读取发送有效数据
#define WTA        0x22    // 向 TX-Address 寄存器写入发送地址
#define RTA        0x23    // 从 TX-Address 寄存器读取发送地址
#define RRP        0x24
// 从 RX-Payload 寄存器读取接收到的有效数据
```

SPI 读函数

步骤一：MISO 线准备好需要发送的数据位

步骤二：SCK 置高，主机读取 MISO 线上的数据

步骤三：SCK 置低，准备接收数据的下一位

以上步骤循环执行 8 次，通过 SPI 从器件上读数据完成！数据的传输时，高位在前，低位在后。

```
unsigned char SpiRead(void)
{
    unsigned char j;
    for (j=0;j<8;j++)
    {
         DATA_BUF=DATA_BUF<<1;
        SCK=1;
        if (MISO)  //读取最高位，保存至最末尾，通过左移位完成整个字节
        {
```

```
            DATA_BUF|=BYTE_BIT0;
        }
        else
        {
            DATA_BUF&=~BYTE_BIT0;
        }
        SCK=0;
    }
    return DATA_BUF;
}
```

SPI 写函数

步骤一：MOSI 线准备好需要发送的数据位

步骤二：SCK 置高，器件读取 MOSI 线上的数据

步骤三：SCK 置低，准备发送数据的下一位

以上步骤循环执行 8 次，通过 SPI 向器件发送数据完成！数据的传输时，低位在前，高位在后。

```
void SpiWrite(unsigned char send)
{
    unsigned char i;
    DATA_BUF=send;
    for (i=0;i<8;i++)
    {
        if (DATA7) //总是发送最高位
        {
            MOSI=1;
        }
        else
        {
            MOSI=0;
        }
        SCK=1;
        DATA_BUF=DATA_BUF<<1;
        SCK=0;
    }
}
```

主机通过 SPI 接口向 905 配置寄存器写入信息

步骤一：CSN 置低电平，SPI 接口开始等待第一条指令；

步骤二：调用 SpiWrite 函数，向 nrf905 发送 WC 指令，准备写入配置信息；

步骤三：反复调用 SpiWrite 函数，向器件配置寄存器写入配置信息；

步骤四：CSN 置高电平，结束 SPI 通讯。即 nRF905 配置完成！

```
void Config905(void)
{
    uchar i;
    CSN=0;                      // CSN 片选信号，SPI 使能
    SpiWrite(WC);               // 向 905 芯片写配置命令
    for (i=0;i<RxTxConf.n;i++)       //  循环写入配置信息
    {
        SpiWrite(RxTxConf.buf[i]);  //RxTxConf 保存预先设置好的配置信
息
    }
    CSN=1;                      // 结束 SPI 数据传输
}
```

nRF905 发送数据程序操作

步骤一：通过 SpiWrite 函数发送 WTP 命令，准备写入 TX 有效数据

步骤二：循环调用 SpiWrite 向 TX-Payload 寄存器写入 TX 有效数据（中间须夹有 CSN 电平变化）

步骤三：延时

步骤四：通过 SpiWrite 函数发送 WTA 命令，准备写入 TX 地址

步骤五：循环调用 SpiWrite 向 TX-Address 寄存器写入 TX 地址

步骤六：TRX_CE=1；开始发送数据，延时，nRF905 数据发送完成，当 nRF905 接收到一条完成的信息时，会将 DR 引脚置高。以下这段代码和范例中提供的有所不同，做了较大的简化，只留下必要的部分

发送状态初始化

```
void SetTxMode(void)
{
    TX_EN=1;
    TRX_CE=0;
    Delay(1);              // delay for mode change(>=650us)
}
```

数据打包后发送

```
void TxPacket(void)
{
    uchar i;
    CSN=0;
    SpiWrite(WTP);                  // Write payload command
    for (i=0;i<32;i++)
    {
        SpiWrite(TxBuf[i]);         // 写入 32 直接发送数据
    }
    CSN=1;                      // 关闭 SPI,保存写入的数据
    Delay(1);
    CSN=0;                      // SPI 使能,准备写入地址信息
    SpiWrite(WTA);                  // 写数据至地址寄存器
    for (i=0;i<4;i++)           // 写入 4 字节地址
    {
        SpiWrite(RxTxConf.buf[i+5]);
    }
    CSN=1;                      // 关闭 SPI
    TRX_CE=1;                   // 进入发送模式,启动射频发送
    Delay(1);                   // 进入 ShockBurst 发送模式后,芯片保证数据
发送完成后返回 STANDBY 模式
    TRX_CE=0;
}
```

联系电话：13704018223 陈 工 E-mail: chj_006@sina.com
在线咨询：QQ:35625400 474882985 MSN:1188mm88@hotmail.com

nRF905 数据接收程序操作

步骤一：TRX_CE=0；必须将此引脚置低，使 905 进入 standby 模式

步骤二：发送 RRP 指令

步骤三：循环调用 SpiRead 函数，读取接收到的数据

步骤四：等待 DR 和 AM 引脚复位为低电平

AM 地址匹配，接收到有效地址，被置高

DR 接收到有效数据包，并解码后，被置高，当所有有效数据被读取后，nRF905 将 AM 和 DR 置低，最后需要注意的是，必须首先设置器件的发送/接收模式才能保证有效的数据发送接收

设置 nrf905 进入接收模式

```
void SetRxMode(void)
{
    TX_EN=0;
    TRX_CE=1;
    Delay(1);              // delay for mode change(>=650us)
}
```

数据收到后读取保存

```
void RxPacket(void)
{
    uchar i;
    TRX_CE=0;                    // 设置 905 进入待机模式
    CSN=0;                       // 使能 SPI
    SpiWrite(RRP);                   // 准备读取接收到的数据
    for (i=0;i<32;i++)
    {
        RxBuf[i]=SpiRead();      // 通过 SPI 接口从 905 芯片读取数据
    }
    CSN=1;                       // 禁用 SPI
    while(DR||AM);
    TRX_CE=1;

}
```

无线应用注意事项

(1) 无线模块的 VCC 电压范围为 1.8V-3.6V 之间，不能在这个区间之外，超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右。

(2) 除电源 VCC 和接地端，其余脚都可以直接和普通的 51 单片机 IO 口直接相连，无需电平转换。当然对 3V 左右的单片机更加适用了。

(3) 硬件上面没有 SPI 的单片机也可以控制本模块，用普通单片机 IO 口模拟 SPI 不需要单片机真正的串口介入，只需要普通的单片机 IO 口就可以了，当然用串口也可以了。模块按照接口提示和母板的逻辑地连接起来

(4) 标准 DIP 插针，如需要其他封装接口，或其他形式的接口，可联系我们定做。

(5) 任何单片机都可实现对无线模块的数据收发控制，并可根据我们提供的程序，然后结合自己擅长的单片机型号进行移植;

(6) 频道的间隔的说明：实际要想 2 个模块同时发射不相互干扰，两者频道间隔应该至少相差 1MHZ，这在组网时必须注意,否则同频比干扰。

(7) 实际用户可能会应用其他自己熟悉的单片机做为主控芯片，所以，建议大家在移植时注意以下 4 点：

A:确保 IO 是输入输出方式，且必须设置成数字 IO;

B:注意与使用的 IO 相关的寄存器设置，尤其是带外部中断、

带 AD 功能的 IO，相关寄存器一定要设置好;

C:调试时先写配置字，然后控制数据收发

D:注意工作模式切换时间

我们的承诺

最后，欢迎您使用我们的产品，在应用中有技术问题请及时向我们联系，我们会予以技术知道，同时运输中出现产品问题我们会全面责任并予以更换。

愿与您一起走向成功

联系电话：13704018223 陈 工 E-mail: chj_006@sina.com
在线咨询：QQ:35625400 474882985 MSN:1188mm88@hotmail.com